Application of Reduced Order Models in Aircraft Gust Response Studies

Stephen P. I. Williams[1], Dorian P. Jones[2], Ann L. Gaitonde[3], Chris Wales[4] and Samantha J. Huntley[5]
Department of Aerospace Engineering, University of Bristol, Bristol, England, BS8 1TR

The aim of this paper is to investigate the role Reduce Order Models (ROMs) can play in the early phases of aircraft design. During such periods of design, resources (both in terms of hardware and time) available for Computational Fluid Dynamics (CFD) are at a premium. As gust response cases, which are often critical load cases, are traditionally computationally expensive, a tool that allows for rapid, accurate modelling of such cases would be of great use. This paper first looks at the initial development of such a ROM, before exploring methods by which their computational cost can be further reduced. Ultimately this led to the development of a ROM that, for an inviscid wing model, requires approximately a tenth of the resources required by full order CFD simulation to carry out one gust case, and near negligible cost to model further cases at the same Mach number.

Nomenclature

$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$	=	system matrices
$\widetilde{\boldsymbol{A}}$, $\widetilde{\boldsymbol{B}}$, $\widetilde{\boldsymbol{C}}$, $\widetilde{\boldsymbol{D}}$	=	discrete-time system matrices
C_l, C_m	=	lift and pitching moment coefficients
$\bar{C}_l$, $\bar{C}_m$	=	change in the lift and pitching moment coefficients from the steady state values
$\boldsymbol{H}(t)$	=	continuous-time impulse response matrix
$\widetilde{\boldsymbol{H}}(t)$	=	discrete-time impulse response matrix
$\boldsymbol{H_{rs}}(k)$	=	Hankel matrix of size r × s at time level k
$\boldsymbol{H_k}$	=	Markov parameter
h_1	=	first order kernel of the system
k	=	time level
l_g	=	gust length
m	=	number of inputs in a multiple input/output system
n	=	rank of the system
p	=	number of outputs in a multiple input/output system
$\boldsymbol{S}$	=	scaling matrix
s	=	distance penetrated into gust
t	=	time
U	=	gust velocity
U_{ds}	=	gust design/peak velocity
$\boldsymbol{U}$, $\boldsymbol{V}$	=	square unitary matrices
u_∞	=	freestream velocity
$v_g(t)$	=	input vector
x	=	state vector
$\dot{x}$	=	state vector differentiated with respect to time
y	=	output vector

[1] Ph.D Researcher, Department of Aerospace Engineering, Queens Building, University Walk, Bristol, England, BS8 1TR.
[2] Senior Lecturer, Department of Aerospace Engineering.
[3] Senior Lecturer, Department of Aerospace Engineering.
[4] Research Associate, Department of Aerospace Engineering.
[5] Research Associate, Department of Aerospace Engineering.

y_1 = non-linear system response to a pulse input of arbitrary
y_{11} = non-linear system response to a pulse input of twice the magnitude as that of y_1
$\bar{y}$ = steady state solution for a fixed v_g
$\boldsymbol{0_m}$ = square identity matrix of size m × m
$\boldsymbol{0_p}$ = square identity matrix of size p × p
Δt = discrete time-step
$\boldsymbol{\Sigma}$ = a diagonal matrix containing non-negative, real numbers
α = angle of attack
δ = flap angle
τ = pseudo time

I. Introduction

Gusts are temporary changes in airflow. In the case of aircraft this is typically taken to be acting in a vertical direction perpendicular to the aircraft motion. CS-25 Regulations [1] require that, in order to be certified, an aircraft must be shown to be able to withstand a range of "1-cosine" shaped gusts (see Fig. 1) with a suitable number of gust lengths within the range of 9m to 107m (some aircraft may need to be subjected to additional gusts for certification). The regulations also specify the peak gust velocities that must be used for any given combination of gust gradients and altitudes.

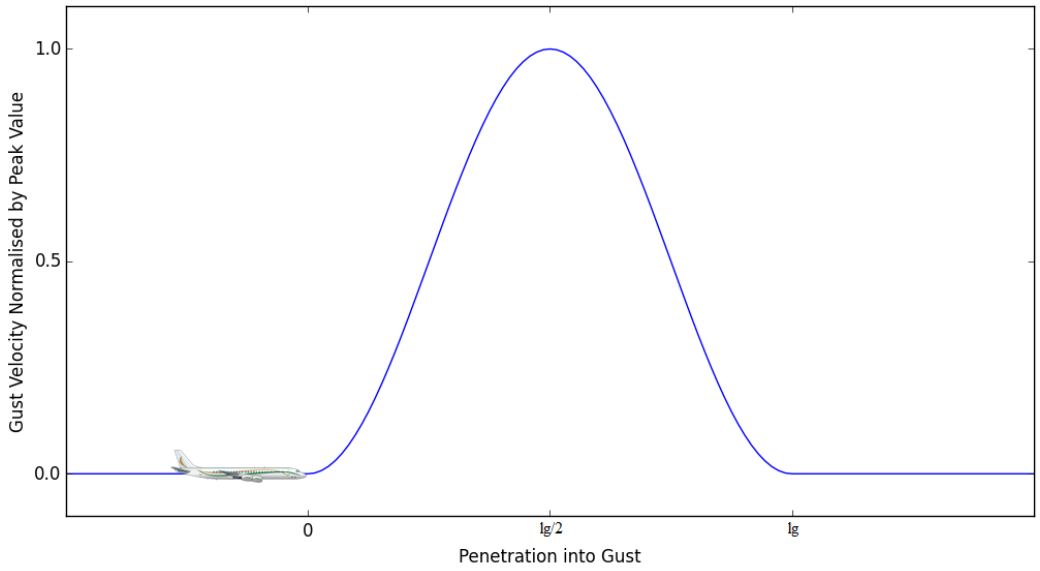

Figure 1. Example of a "1-cosine" gust; with distance penetrated into gust given in terms of number of gust lengths (l_g).

It is universally accepted that aerodynamic gust cases present some of the most critical design cases both in terms of loads and fatigue [2]. As such they are a critical component of any design process. However, despite their importance they represent an area where large improvements can still be made. This comes as a result of their complexity to model, both in wind tunnels and within Computational Fluid Dynamics (CFD) modelling. For the former this is as a result of their special setup requirements within the wind tunnel to generate any form of gust, including the "1-cosine" gust profile [3] which, as earlier discussed, is required for certification. For the latter it is as a result of the necessity of a fine mesh across the domain to prevent the gust being dissipated when using standard techniques; thus making the computational method prohibitively expensive [4]. It is for this reason that historically, simpler mathematical models of gusts have been used, but the required simplifying assumptions limit the applicability of such methods.

This paper looks at the initial development of a CFD based Reduced Order Model (ROM) capable of rapid modelling of an aircraft's response to the aforementioned "1-cosine" gusts, with an emphasis on those covered within CS-25 regulations. The purpose of this ROM is to allow for reliable identification of critical gust loads early

within the loads design process where computational cost is important. These critical cases could then be further explored with the more expensive, and thus more limited, computational and/or wind tunnel resources available to designers. This would help to improve the underlying structure of the aircraft (particularly the wings) and thus reduce weight (and ergo fuel) by being less conservative than is currently the case. Some reduced order models have already been developed such as by Zaide and Raveh [5] and Wales *et al* [4]. However this paper looks closer at the computational requirements of ROMs and how this requirement can be decreased; making ROMs a more promising tool.

II. Mathematical Background of the Baseline Reduced Order Model

The baseline ROM was built using the methods explored by Wales *et al* [4]. This base model ROM allowed for the modification and adaptation of Wales' work; with the aim of reducing the computational cost incurred by the ROM, without sacrificing the accuracy. For completeness his method is summarised below.

A. Eigensystem Realisation Algorithm for Continuous Systems

If changes to the force coefficients (such as the lift and pitching moment coefficients; which are the two this paper considers) from the non-linear state flow can be assumed to be close to linear, then for a continuous time system, the behaviour of said system can be expressed as:

$$\begin{aligned} \dot{x}(t) &= \boldsymbol{A}x(t) + \boldsymbol{B}v_g(t) \\ y(t) &= \boldsymbol{C}x(t) + \boldsymbol{D}v_g(t) \end{aligned} \tag{1}$$

Where $v_g(t)$ is the input vector and:

$$y = \begin{bmatrix} \hat{C}_l \\ \hat{C}_m \end{bmatrix} \tag{2}$$

If Eq. (1) is taken and multiplied through by $\boldsymbol{\Phi}(t) = e^{-\boldsymbol{A}t}$, then the differential equation can be solved as follows [6]:

$$\begin{aligned} x(t) &= \boldsymbol{\Phi}(\mathrm{t})x(0) + \int_0^t \boldsymbol{\Phi}(t-\tau)\mathbf{B}\mathrm{v_g}(t)d\tau \\ y(t) &= \boldsymbol{C}\boldsymbol{\Phi}(t)x(0) + \int_0^t \boldsymbol{C}\boldsymbol{\Phi}(t-\tau)\boldsymbol{B}v_g(\tau)d\tau + \boldsymbol{D}v_g(t) \end{aligned} \tag{3}$$

When $v_g = 0$ the first term set of terms (on the right hand side) of the first equation is the system response and is known as the free response. The second set of terms is the forced response and is the system response to the input vector for the zero initial state. In the current application, $x(0) = 0$, therefore the total response of the system is given by:

$$\begin{aligned} x(t) &= \int_0^t \boldsymbol{\Phi}(t-\tau)\boldsymbol{B}v_g(\tau)d\tau \\ y(t) &= \int_0^t (\boldsymbol{C}\boldsymbol{\Phi}(t-\tau)\boldsymbol{B} + \boldsymbol{D}\delta(\mathrm{t}-\tau))\mathrm{v_g}(\tau)d\tau \end{aligned} \tag{4}$$

If we introduce the continuous-time impulse response matrix $\boldsymbol{H}(t)$, which contains the system output to a unit impulse for gust velocity, then $y(t)$ can be rewrote as:

$$y(t) = \int_0^t \boldsymbol{H}(t-\tau) v_g(\tau) d\tau \tag{5}$$

If this continuous-time impulse response matrix could be identified then the Hankel matrix could be formed and thus a continuous ROM could be generated via the Eigensystem Realisation Algorithm (ERA) [7]. However, the majority of CFD codes/solvers work in discretised time rather than continuous. Thus it is necessary to consider this method in discrete time.

B. Eigensystem Realisation Algorithm for Discrete Systems

The Markov parameters are functions of the discrete system matrices. To identify them a first order implicit finite difference scheme was used for the time derivative of the current scheme. These Markov parameters and the input are the only pieces of data necessary to recreate the response of the discrete linear system for any input. The discrete form of Eq. (1) is:

$$\begin{aligned} \frac{\tilde{x}_k - \tilde{x}_{k-1}}{\Delta t} &= \boldsymbol{A}\tilde{x}_k + \boldsymbol{B}\left(\tilde{v}_g\right)_k \\ \tilde{y}_k &= \boldsymbol{C}\tilde{x}_k + \boldsymbol{D}(\tilde{v}_g)_k \end{aligned} \tag{6}$$

which can be rearranged to obtain:

$$\begin{aligned} \tilde{x}_k &= \widetilde{\boldsymbol{A}}\tilde{x}_{k-1} + \widetilde{\boldsymbol{B}}\left(\tilde{v}_g\right)_k \\ \tilde{y}_k &= \widetilde{\boldsymbol{C}}\tilde{x}_k + \widetilde{\boldsymbol{D}}\left(\tilde{v}_g\right)_k \end{aligned} \tag{7}$$

The discrete system matrices are given by:

$$\begin{aligned} \widetilde{\boldsymbol{A}} &= (I - \boldsymbol{A}\Delta t)^{-1} \\ \widetilde{\boldsymbol{B}} &= (I - \boldsymbol{A}\Delta t)^{-1}\boldsymbol{B}\Delta t \\ \widetilde{\boldsymbol{C}} &= \boldsymbol{C} \\ \widetilde{\boldsymbol{D}} &= \boldsymbol{D} \end{aligned} \tag{8}$$

In most instances the system matrix $\boldsymbol{D}$ is both small and known (for gust responses it is simply zero), thus the output vector (Eq. (6)) can be simplified to $\tilde{y}^m(k) = \boldsymbol{C}\tilde{x}_k$.
If a one-sided z-transform is applied to the output vector, then the solution of this equation is:

$$\tilde{y}^m(k) = \sum_{n=0}^{k} \widetilde{\boldsymbol{H}}(k-n)\left(\tilde{v}_g\right)_n \tag{9}$$

Here $\widetilde{\boldsymbol{H}}$ is the discrete-time impulse response matrix. The columns of this matrix are the outputs for a unit impulse input on each input separately. Should the initial system be undisturbed (i.e. if $\tilde{x}_{-1} = 0$) then the forced response of the system is given by:

$$\begin{aligned}\tilde{y}_0^m &= \tilde{\boldsymbol{C}}\tilde{\boldsymbol{B}}\left(\tilde{v}_g\right)_0 \\ \tilde{y}_1^m &= \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}}\left(\tilde{v}_g\right)_0 + \tilde{\boldsymbol{C}}\tilde{\boldsymbol{B}}\left(\tilde{v}_g\right)_1 \\ \tilde{y}_2^m &= \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^2\tilde{\boldsymbol{B}}\left(\tilde{v}_g\right)_2 + \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}}\left(\tilde{v}_g\right)_1 + \tilde{\boldsymbol{C}}\tilde{\boldsymbol{B}}\left(\tilde{v}_g\right)_2\end{aligned} \tag{10}$$

This can be written in matrix form as:

$$\tilde{y}_l^m = [\boldsymbol{H_l}, \quad \boldsymbol{H_{l-1}}, \quad \ldots, \quad \boldsymbol{H_2}, \quad \boldsymbol{H_1}, \quad \boldsymbol{H_0}] \begin{bmatrix} \left(\tilde{v}_g\right)_0 \\ \left(\tilde{v}_g\right)_1 \\ \vdots \\ \left(\tilde{v}_g\right)_{l-2} \\ \left(\tilde{v}_g\right)_{l-1} \\ \left(\tilde{v}_g\right)_l \end{bmatrix} \tag{11}$$

Where the following sequence $\boldsymbol{H}_k,\ k = 0,\ \infty$:

$$\{\boldsymbol{H}_0, \quad \boldsymbol{H_1}, \quad \ldots \quad \boldsymbol{H}_k, \quad \ldots\} = \{\tilde{\boldsymbol{C}}\tilde{\boldsymbol{B}}, \quad \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}}, \quad \ldots, \quad \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^k\tilde{\boldsymbol{B}}, \quad \ldots\} \tag{12}$$

is called the impulse-response, or the weighting sequence and is also the Markov sequence of the system.

C. Pulse Extraction

Using a dynamically linearised Euler code the impulse-response sequence can be generated. Alternatively, Volterra theory can be used to approximately identify the linear responses of a system from the non-linear CFD code [8–10]. A truncated Volterra series for small inputs can be used to represent the solution for weakly non-linear systems. Should the system be able to be approximated by a second order Volterra series, then the first order kernel of the system can be identified as:

$$h_1 = 2y_1 - \frac{1}{2}y_{11} \tag{13}$$

h_1 captures some level of the amplitude dependence for non-linear systems, thus is generally different from the purely linear pulse response. However, should the non-linear system exhibit near linear behavior in response to small inputs, then it can be assumed that the non-linear first order kernel is that of an approximating linear system. Ergo the linear responses can be found from two response of the non-linear code.

D. System Reduction

The Markov sequence and the system input uniquely determine the forced response of a linear system. Ergo, if two systems have the same Markov sequence, then they will produce the same forced response for a given input. It is worth noting that a system of rank n has its forced system response exactly defined by $2n$ Markov parameters [4].

Using ERA it is possible to identify the system matrices, assuming the relevant Hankel matrices can be constructed. However for the current large system, which is already known, there is no requirement to identify the full order system matrices. The requirement is rather to capture the dominant behavior of the system. This reduces the size of the require Hankel matrix required. The Hankel matrix is given by:

$$\boldsymbol{H}_{rs}(k) = \begin{bmatrix} \boldsymbol{H}_k & \boldsymbol{H}_{k+1} & \cdots & \boldsymbol{H}_{k+s-1} \\ \boldsymbol{H}_{k+1} & \boldsymbol{H}_{k+2} & \cdots & \boldsymbol{H}_{k+s} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{H}_{k+r-1} & \boldsymbol{H}_{k+r} & \cdots & \boldsymbol{H}_{k+s+r-2} \end{bmatrix} \tag{14}$$

In $\boldsymbol{H}_k$ each column contains the outputs for a unit pulse input to a single channel at time level k (all other inputs are set to zero). For a system with m inputs and p outputs, each Markov parameter is a $p \times m$ matrix. Therefore the size of the Hankel matrix is $rp \times sm$.

The Hankel matrix can be used to identify reduced system matrices via ERA. The first step of the basic ERA [7] method is a singular-value decomposition (SVD) of $\boldsymbol{H}_{rs}(k)$ when $k = 0$ which is given by:

$$\boldsymbol{H}_{rs}(0) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{15}$$

The elements of $\boldsymbol{\Sigma}$ are sorted in descending order, with the largest singular values first. The Hankel matrix is then partitioned as:

$$\boldsymbol{H}_{rs}(0) = [\boldsymbol{U}_r \quad \boldsymbol{U}_0] \begin{bmatrix} \boldsymbol{\Sigma}_r & 0 \\ 0 & \boldsymbol{\Sigma}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_r \\ \boldsymbol{V}_0 \end{bmatrix}^T \tag{16}$$

$\boldsymbol{H}_{rs}(0)$ can be approximated by:

$$\boldsymbol{H}_{rs}(0) = \boldsymbol{U}_r \boldsymbol{\Sigma}_{\mathrm{r}} \boldsymbol{V}r^T \tag{17}$$

There are several possible realisations and in this work a balanced realisation is used given by [7]:

$$\begin{aligned} \widetilde{\boldsymbol{A}}_r &= \boldsymbol{\Sigma}_{\mathrm{r}}^{-\frac{1}{2}} \, \boldsymbol{U}_r^T \boldsymbol{H}_{rs}(1) \boldsymbol{V}_r \boldsymbol{\Sigma}_{\mathrm{r}}^{-\frac{1}{2}} \\ \widetilde{\boldsymbol{B}}_r &= \boldsymbol{\Sigma}_{\mathrm{r}}^{\frac{1}{2}} \, \boldsymbol{V}_r^T \boldsymbol{E}_m \\ \widetilde{\boldsymbol{C}}_r &= \boldsymbol{E}_p^T \boldsymbol{U}_r \boldsymbol{\Sigma}_{\mathrm{r}}^{\frac{1}{2}} \end{aligned} \tag{18}$$

Where:

$$\begin{aligned} \boldsymbol{E}_{\boldsymbol{m}}^T &= [\mathbf{I}_{\mathrm{m}}, \quad \mathbf{0}_{\mathrm{m}}, \quad \mathbf{0}_{\mathrm{m}}, \quad \ldots, \quad \mathbf{0}_{\mathrm{m}}] \\ \boldsymbol{E}_p^T &= [\mathbf{I}_{\mathrm{p}}, \quad \mathbf{0}_{\mathrm{p}}, \quad \mathbf{0}_{\mathrm{p}}, \quad \ldots, \quad \mathbf{0}_p] \end{aligned} \tag{19}$$

Where $\mathbf{I_m}$ and $\mathbf{I_p}$ are identity matrices of size m×m and p×p respectively, and $\mathbf{0_m}$ and $\mathbf{0_p}$ are zero matrices of size m×m and p×p respectively.

E. Steady State Correction

For some cases, the ROM may be improved by introducing steady state correcting. This is given by:

$$\begin{aligned} \tilde{x}_k &= \boldsymbol{A}\tilde{x}_{k-1} + \boldsymbol{B}\left(\tilde{v}_g\right)_k \\ \tilde{y}_k &= \boldsymbol{S}\left(\tilde{v}_g\right)_k \widetilde{\boldsymbol{C}}\tilde{x}_k \end{aligned} \tag{20}$$

Where the scaling matrix can be calculated as:

$$\boldsymbol{S}\left(\tilde{v}_g\right) = \frac{\bar{y}\left(\tilde{v}_g\right)_{CFD}}{\bar{y}\left(\tilde{v}_g\right)_{Linear}} \tag{21}$$

and $\bar{y}$ is defined as:

$$\bar{y} = \begin{bmatrix} \bar{C}_l \\ \bar{C}_m \end{bmatrix}_{Steady} \tag{22}$$

III. Adaptations to the Baseline ROM

Once the baseline ROM was built and tested (see section IV.B.) it was modified progressively to look at how changes could be made that reduce the computational cost without negatively impacting upon the accuracy of the results. These changes are outlined below.

A. Single Sharp-Edged Gust

The first modification to the baseline ROM was to remove the necessity for running two sharp-edged gusts. The baseline ROM required two sharp-edged gusts to be run, one with twice the peak gust velocity of the other, to produce the pulse response (see Eq. (13)). If one of the sharp-edged gusts has a magnitude that is sufficiently small, then it can be assumed that a sharp-edged gust with half the magnitude is approximately equal to the steady state conditions. Thus the pulse extraction method detailed earlier can be simplified.

B. Variable Time-Step

Originally, the ROMs (both the baseline and single sharp-edged gust) were built using sharp-edged gusts that had a constant time step. This time step was necessarily small in order to ensure the results of the CFD were suitably accurate. However, this is only required when the gust is interacting with the model. Therefore, having large time steps to bring the gust into the computational domain, before transitioning to smaller time steps for the remainder of the simulation, could offer significant savings, without having a notable detrimental impact on the accuracy of the ROM.

However, before the modification could be fully tested an investigation into the effect that the location of the time-step transition point has on the results was required. Whilst most of system response would be captured when the gust first impacts the model, there may be some parts of the response that are generated shortly before this point. To investigate this issue, multiple sharp edged gust simulations were set up for an arbitrary gust case. For each one the transition point was located in a different location (measured in terms of mean aerodynamic chord lengths upstream from the leading edge). In addition, all of the simulations were stopped earlier than previous sharp edged gusts in order to reduce the computational resources required to carry out the study. For comparison, two control simulations were also run. Both controls used the constant small time steps previously used, with one ending at the same point as the test cases, and the other running to the old stop point, which allowed for quick verification that the early end of the simulation didn't have a negative impact on the results.

The results of this investigation can be found below in Fig. 2 and 3:

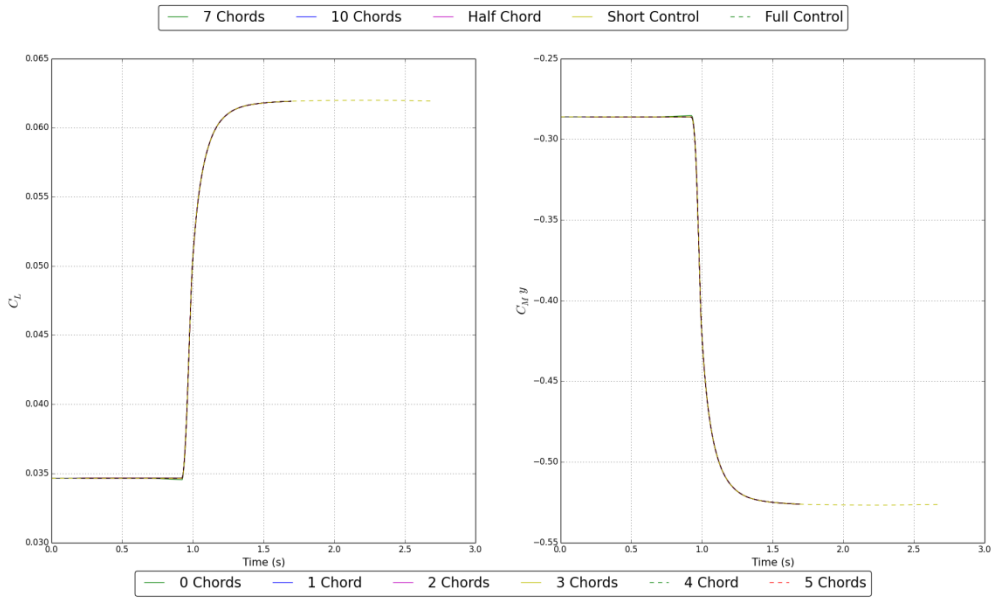

Figure 2. System response to a sharp-edged gust with the time-step size transitioning at different, expressed as the number of mean aerodynamic chord lengths ahead of the leading edge.

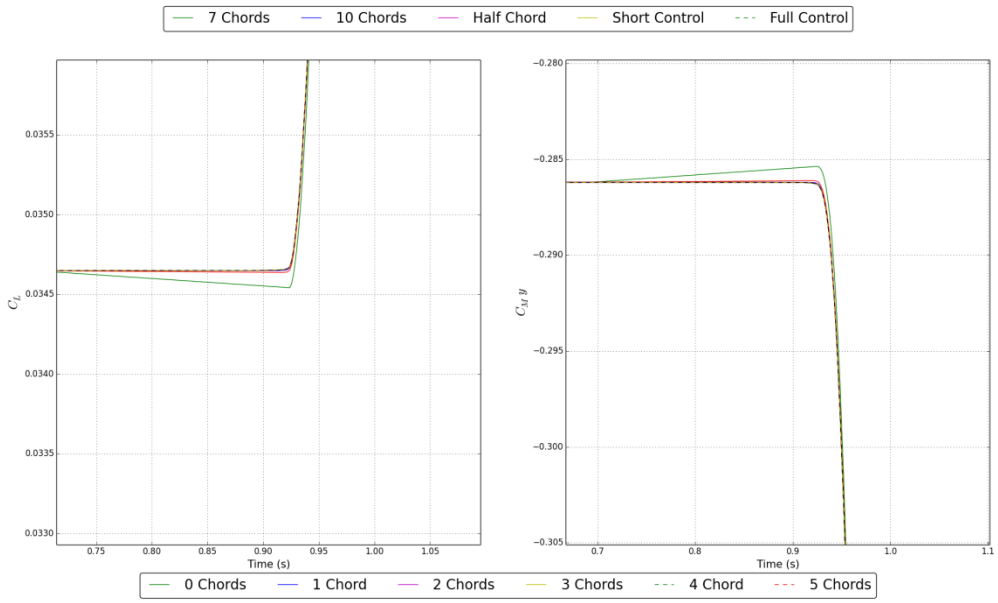

Figure 3. Close-up view of the point of impact for the system response to a sharp-edged gust with the time-step size transitioning at different, expressed as the number of mean aerodynamic chord lengths ahead of the leading edge.

It can be seen that the location of the transition point doesn't greatly affect the system response. However, when the transition occurs on the leading edge itself, a slight degradation in accuracy can be noticed during the initial deviation from the system's steady state. Therefore, it was decided that transitioning the time step one chord ahead of the leading edge would be appropriate; as this accurately captures the system response, whilst also keeping computational costs down. Thus, this single sharp-edged gust based ROM was modified to transition from large time-steps to smaller ones one mean aerodynamic chord length ahead of the leading edge.

C. Reduction in Sharp-Edged Gust End Point

The final modification to the ROM was by altering for how long the sharp-edged gust simulation was run for. Most of the dominant behavior of the system would be expected to be captured by the time the edge of the gust has cleared the model. However, there may be some aspects of the system response that require the gust to travel further downstream in order to be accurately captured. Thus another investigation was carried out to look at the effect that ending the sharp-edged gust simulation at different locations has on the result of the ROM.

To carry out this investigation the single sharp-edged gust based ROM was run for an arbitrary "1-cosine" gust case. Then the result of the sharp-edge gust was manually trimmed to imitate the effect of the simulation being ended early. The end point of the sharp-edged gust simulation was expressed as the number of mean aerodynamic chord lengths past the aft most point of the model the edge of the sharp-edge traversed to before being stopped. The ROM was then run with these modified sharp-edge gust results to see how they affected the output of said ROM.
The results of this investigation can be seen in Fig. 4 below:

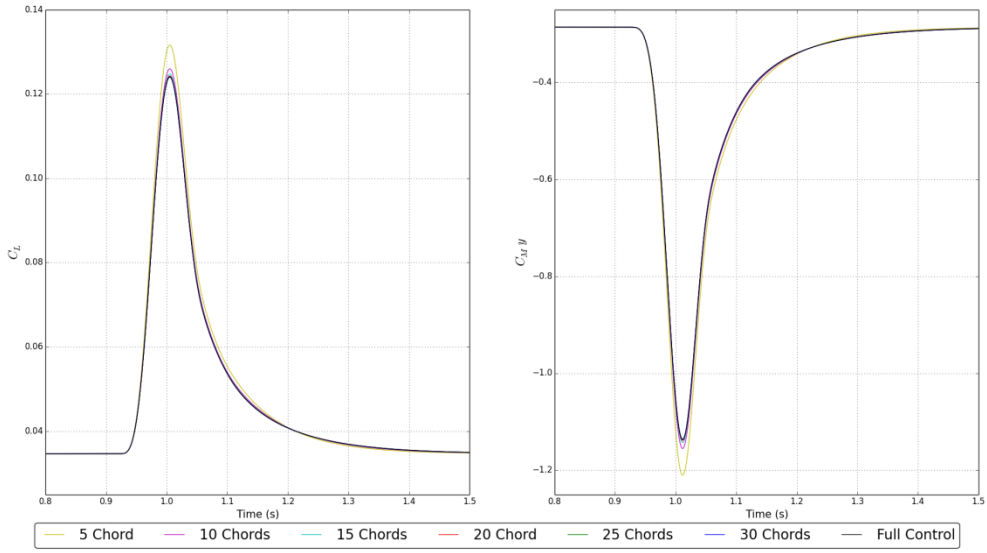

Figure 4. System response of the single sharp-edged gust based ROM when terminating the sharp-edged gust simulation at different points, expressed as the number of mean aerodynamic chords past the aft most point of the wing.

Unexpectedly, the results suggest the gust needs to travel in the order of 30 mean aerodynamic chord lengths downstream after clearing the model for all the dominant behaviors to be accurately captured. However, this distance is far greater than expected.

After examining the ROM it was found that these unexpected results were a result of the way in which the linear gradients (used within the steady state correction) were calculated. The original method for calculating the aforementioned was to take the difference between the final and starting values of the force coefficients within the small magnitude sharp-edged gust and divide by the magnitude of said gust (this was often 1 m/s thus further simplifying the process). However, when ending the sharp-edged gust simulation early, the response has not had enough time to settle, thus altering the final values of the force coefficients (and thus invalidating the steady state corrections).

To overcome this problem a new method of calculating these linear gradients was implemented. From the steady simulations, two different angles of attack (typically 0 and 0.5 degrees) can be used to find the force coefficients for two effective gusts as per Eq. (23) below:

$$U = u_\infty \tan(\alpha) \tag{23}$$

The linear change in the force coefficients between these effective gust velocities can then be calculated. Using this updated method the investigation was repeated giving the result in Fig. 5 below:

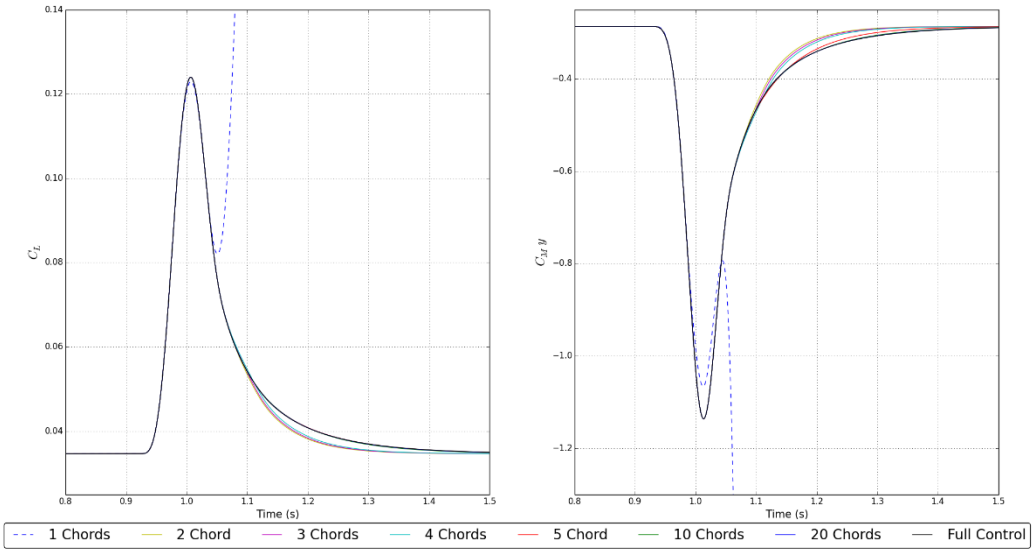

Figure 5. System response, for a short gust, of the single sharp-edged gust based ROM, with updated linear gradient calculation method, when terminating the sharp-edged gust simulation at different points, expressed as the number of mean aerodynamic chords past the aft most point of the wing.

This was then repeated for a larger gust case to ensure the selected cut off point was suitable for a wide range of gust cases. The results of which can be seen in Fig. 6 below:

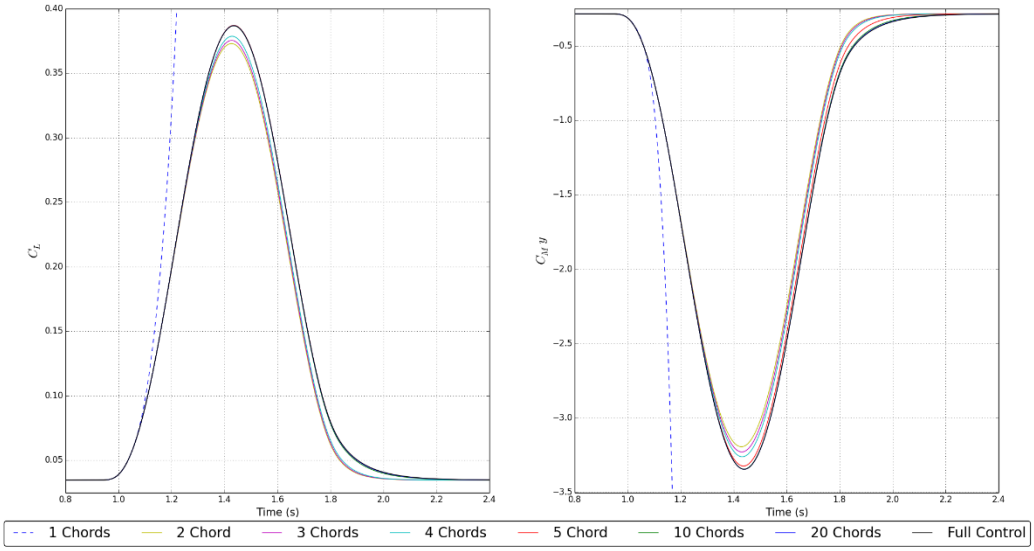

Figure 6. System response, for a long gust, of the single sharp-edged gust based ROM, with updated linear gradient calculation method, when terminating the sharp-edged gust simulation at different points, expressed as the number of mean aerodynamic chords past the aft most point of the wing.

It can be seen that the new method for applying the steady state correction results in ROM outputs much closer to those expected. When the sharp-edge gust data is cut shortly after the gust has cleared the aft most point of the model the result is instability in the response after the initial peak. After this there is a period where the result is stable but suffers in accuracy. Therefore the ROM was modified to end the single sharp-edged gust simulation once the gust has cleared five mean aerodynamic chords past the aft most point of the model. This was done twice, once

without the variable time-steps (i.e. a modified version of the original single sharp-edged gust based ROM) and once with (i.e. a modified version of the variable time-step based ROM).

IV. Results

In order to ascertain the accuracy of the aforementioned methods, the response of the Future Fast Aeroelastic Simulation Technologies (FFAST) wing to various gust cases as calculated by the ROM was compared to the full order CFD response for that gust case. All the gusts investigated were "1-cosine" vertical gusts, as defined by the CS-25 regulations [1] as:

$$U = \begin{Bmatrix} 0, & s < 0 \quad s > l_g \\ \frac{U_{ds}}{2}\left[1 - cos\frac{2\pi s}{l_g}\right], & l_g \geq s \geq 0 \end{Bmatrix} \tag{24}$$

A. Gust Cases

To adequately test the method, four gust lengths were investigated. For all cases the wing was set at zero degrees incidence and the free stream conditions were set as standard sea level, having a velocity of 250 ms^{-1} and thus producing a Mach number of 0.735. With these flight conditions, the gust gradients of 9 m, 40 m, 73 m and 107m were selected (thus satisfying the CS-25 required range of gust lengths). For these gust gradients, the CS-25 specified gust magnitude was used; using no gust alleviation factor and assuming the fight point is not a diving case, for which there are additional regulations that affect the gust magnitude. For all versions of the ROM detailed in this report, steady state correction was applied. The details of the four cases are found in Table 1 below:

Table 1. Specifications of the four gust cases.

Case number	Gust length (m)	Gust magnitude (m/s)	Equivalent angle of attack (deg.)
1	18	9.274	2.124
2	80	11.891	2.723
3	146	13.145	3.010
4	214	14.010	3.208

B. Baseline ROM

Using the four test cases, the baseline ROM was built using the process laid out by Wales *et al* [4] and covered in section II of this paper. The results (see Fig. 7 to Fig. 10) show very close matches to the full order CFD simulations. Thus, the decision to focus on reducing the computational cost of the ROM rather than further improving the accuracy is a valid one.

It should however, be noted that even this baseline ROM offers significant computational savings compared with the full order simulations. To produce a ROM model with this method, the computational resources are approximately equal to running one full order CFD simulation. However, the true savings from the ROM come when modelling multiple gust cases for a given Mach number. The ROM model is valid for any flight conditions with that Mach number, and further responses can be produced with negligible computational cost. This offers potentially very large savings when compared to full order CFD simulations when multiple flight points need to be studied.

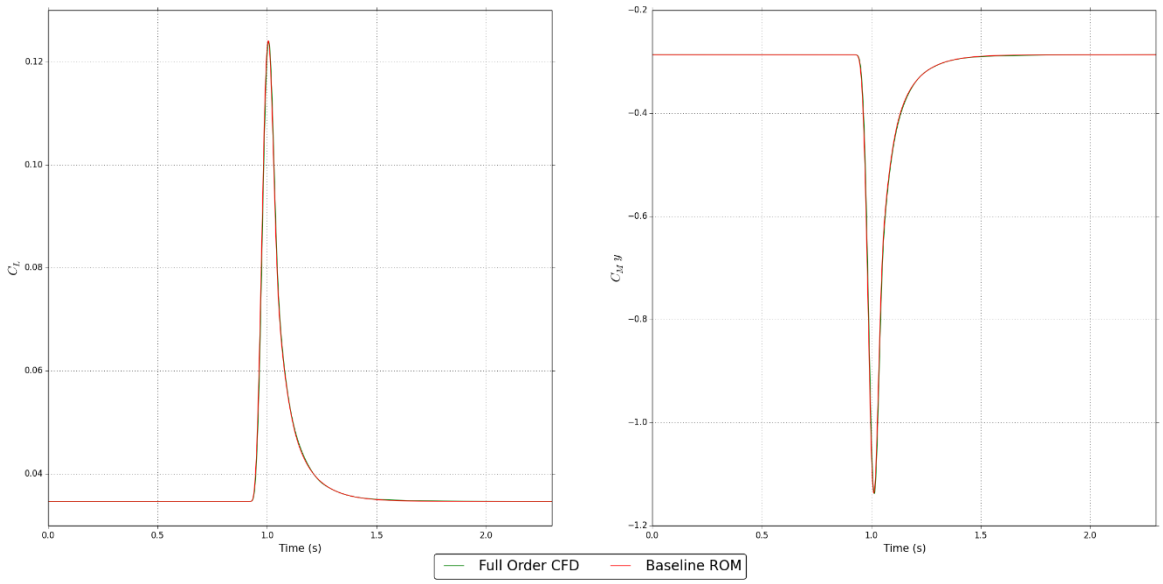

Figure 7. System response for gust case 1 calculated by the baseline ROM.

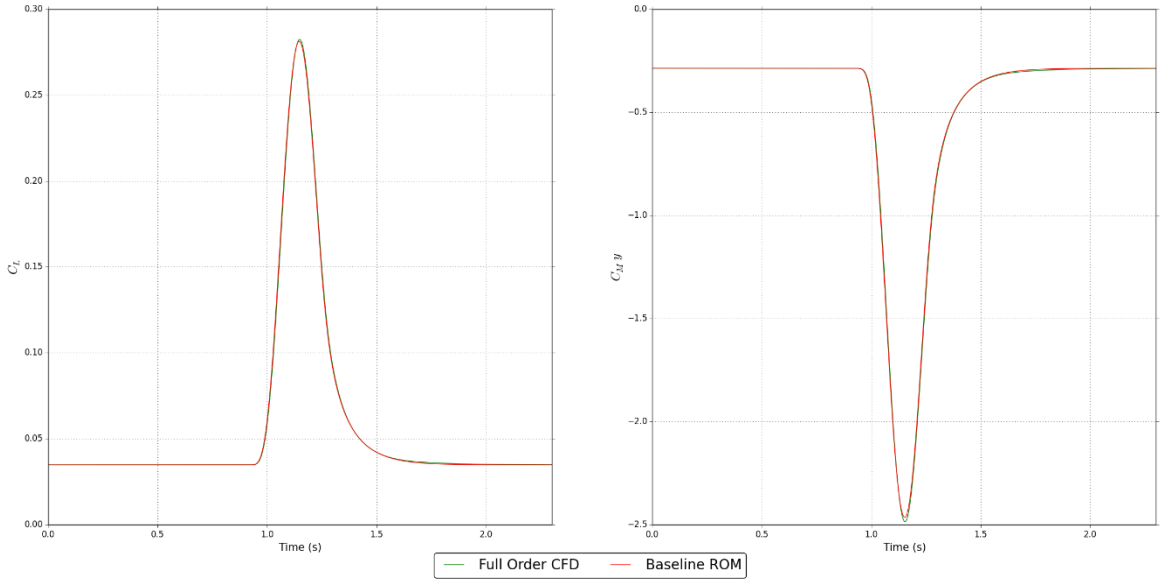

Figure 8. System response for gust case 2 calculated by the baseline ROM.

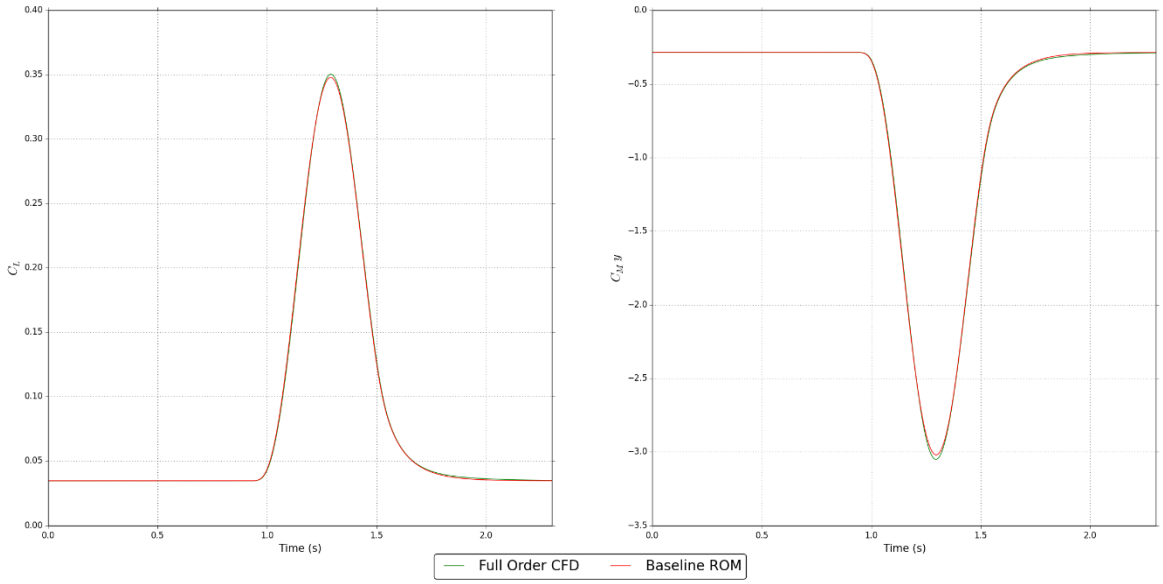

Figure 9. System response for gust case 3 calculated by the baseline ROM.

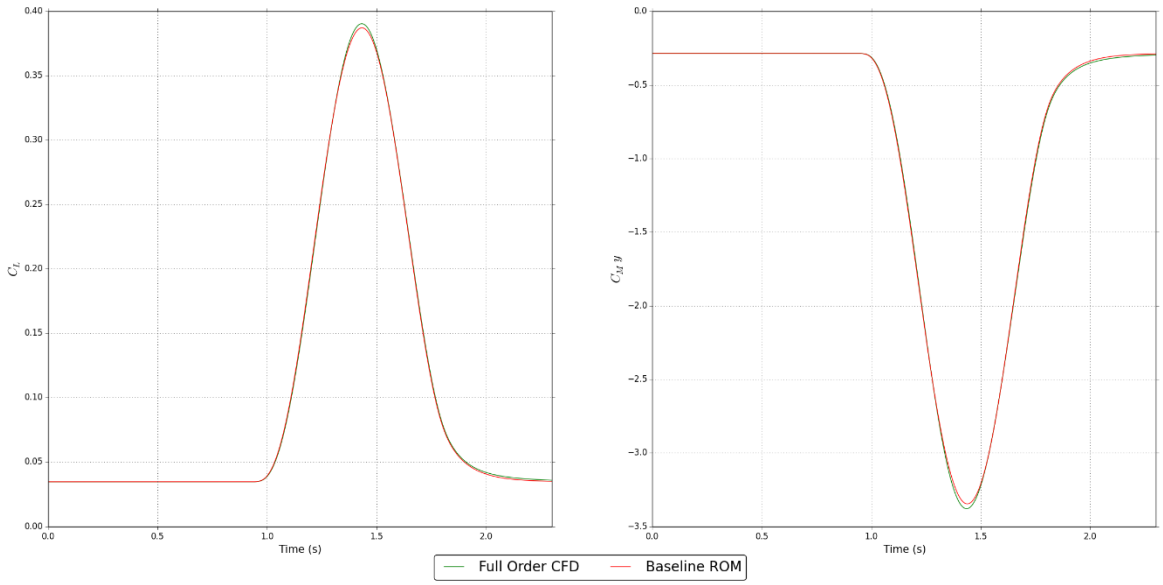

Figure 10. System response for gust case 4 calculated by the baseline ROM.

C. Single Sharp-Edged Gust ROM

The results for the single sharp-edged gust based ROM (shown in Fig. 11 to Fig. 14) are extremely similar to those for the baseline ROM (Fig. 7 to Fig. 10). This shows that the theory detailed earlier in the paper with respect to reducing the number of sharp-edged gusts to one is valid and produces results with no notable loss in accuracy (compared to the baseline ROM). However, as a result of only using one sharp edged gust, the computation cost of building the ROM is greatly reduced; approximately yielding a 45% reduction when compared to the base ROM.

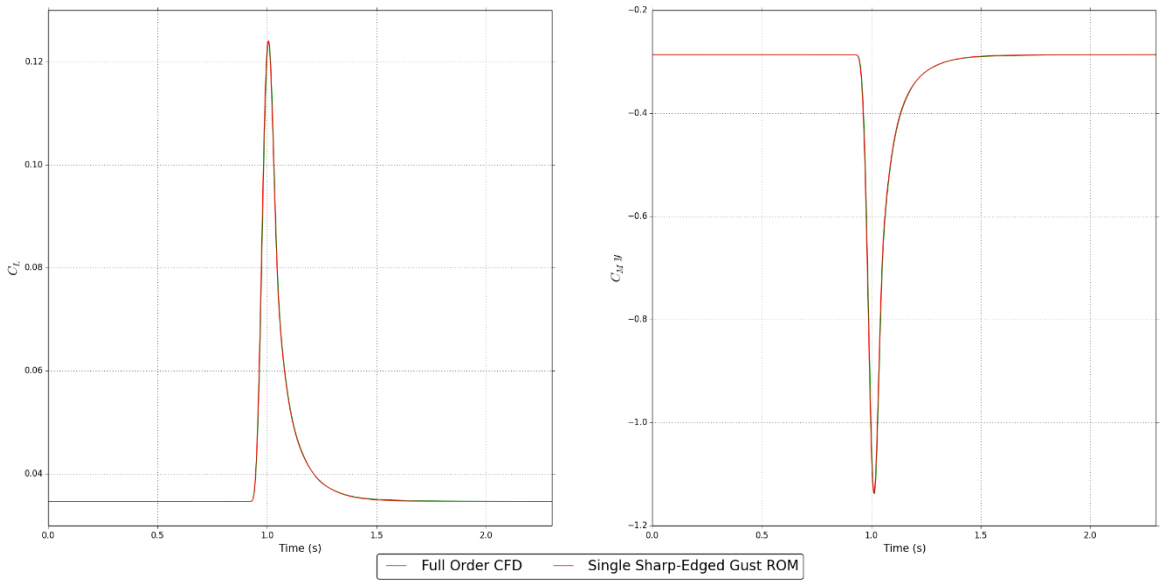

Figure 11. System response for gust case 1 calculated by the single sharp-edged gust based ROM.

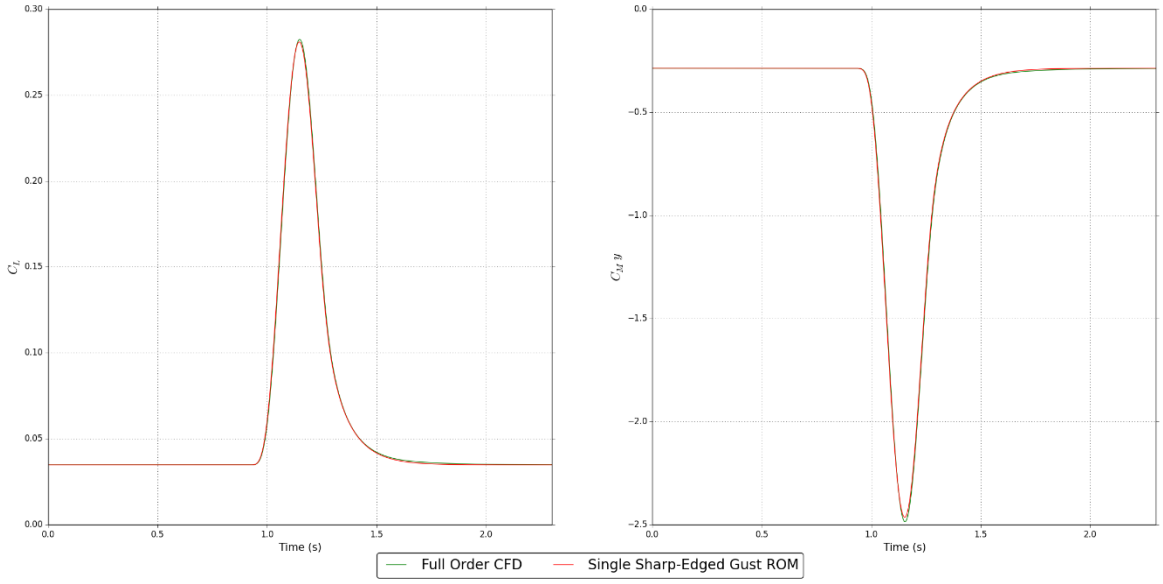

Figure 12. System response for gust case 2 calculated by the single sharp-edged gust based ROM.

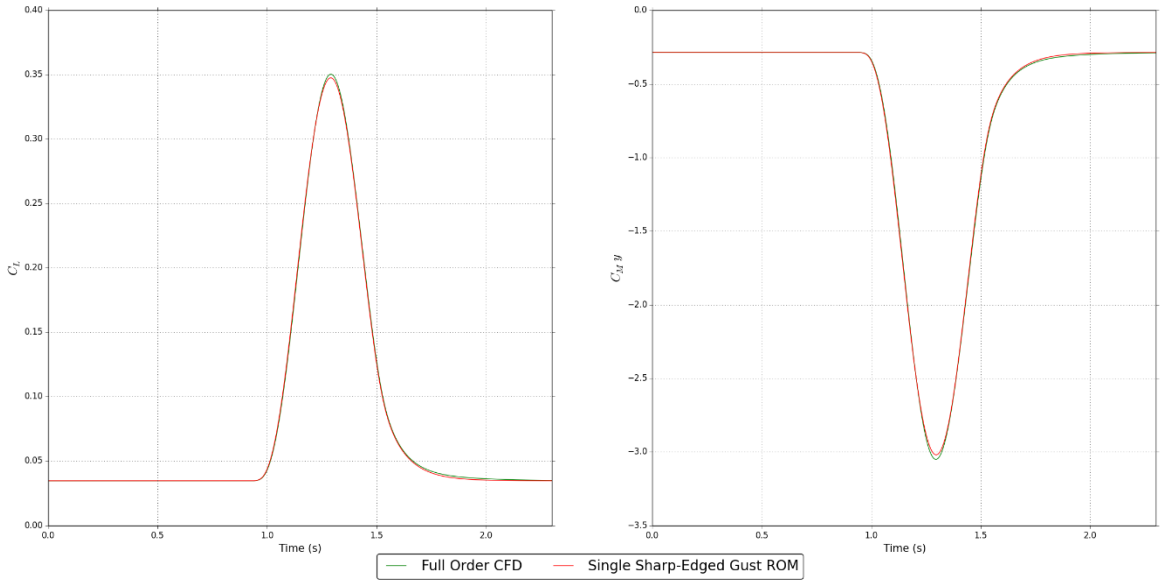

Figure 13. System response for gust case 3 calculated by the single sharp-edged gust based ROM.

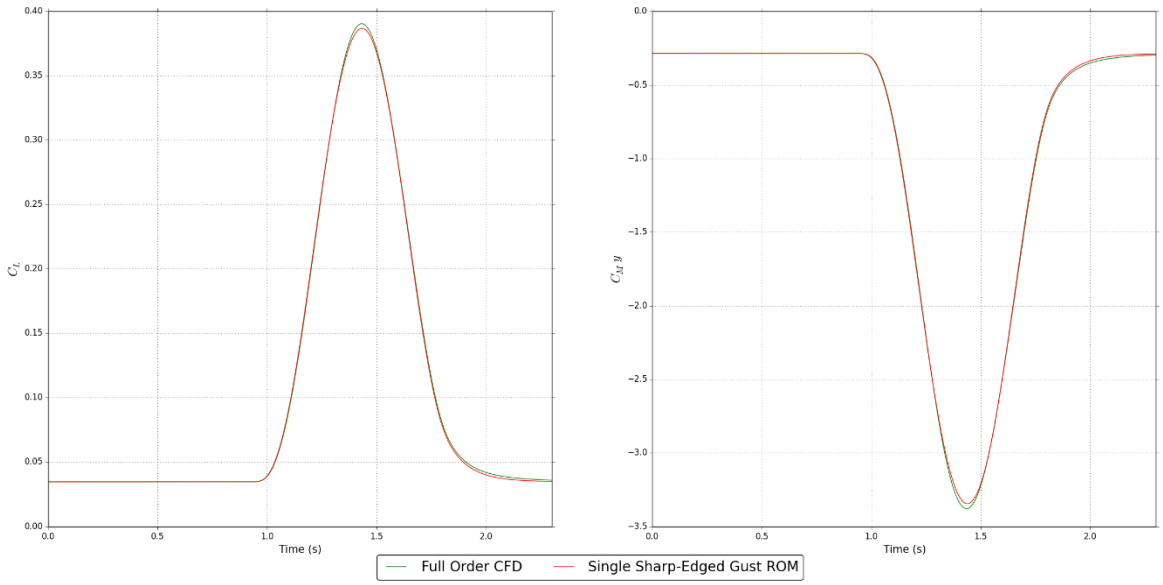

Figure 14. System response for gust case 4 calculated by the single sharp-edged gust based ROM.

D. Variable Time-step

As with the previous modification, it can be seen in Fig. 15 to Fig. 18 that the accuracy of the ROM has been maintained when introducing variable time-steps (to a single sharp-edged gust based ROM); with negligible change in the results produced with respect to the baseline ROM. By varying the time-step size it was possible to reduce the total number of time-steps in the sharp-edged gust simulation from 1750 to 1174. This resulted in a reduction in computational cost of approximately 30% when compared to the single sharp-edged gust based ROM and approximately 60% when compared with the baseline ROM.

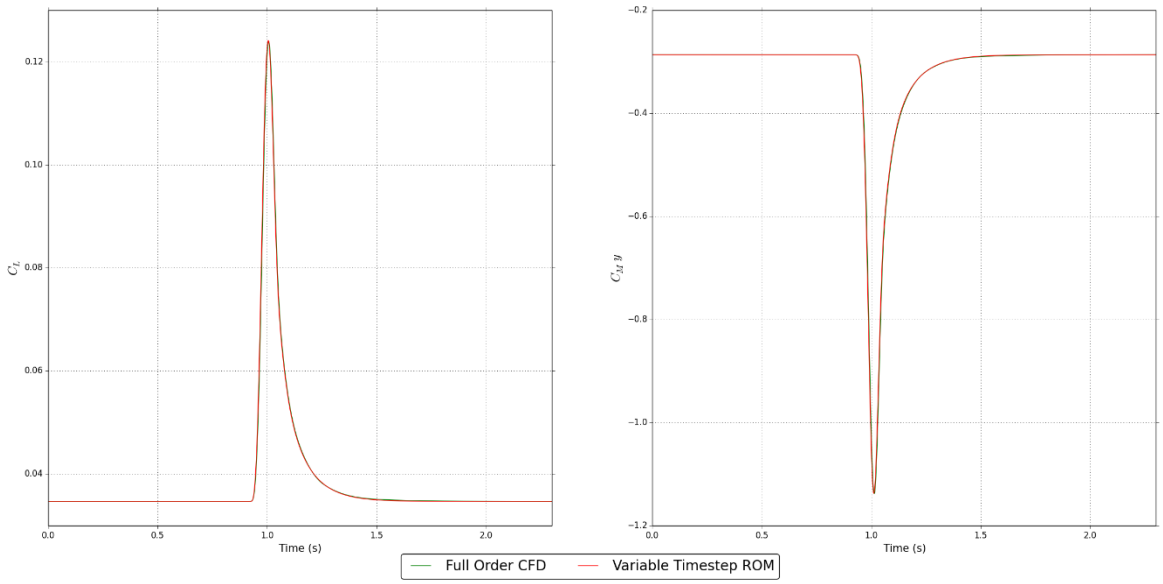

Figure 15. System response for gust case 1 calculated by the variable time-step based ROM.

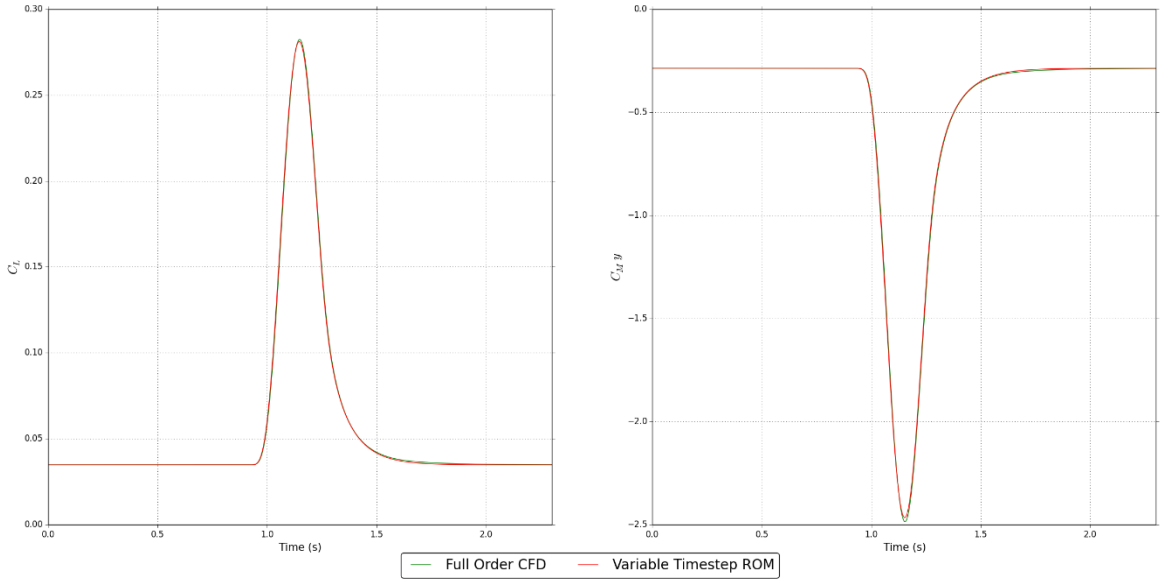

Figure 16. System response for gust case 2 calculated by the variable time-step based ROM.

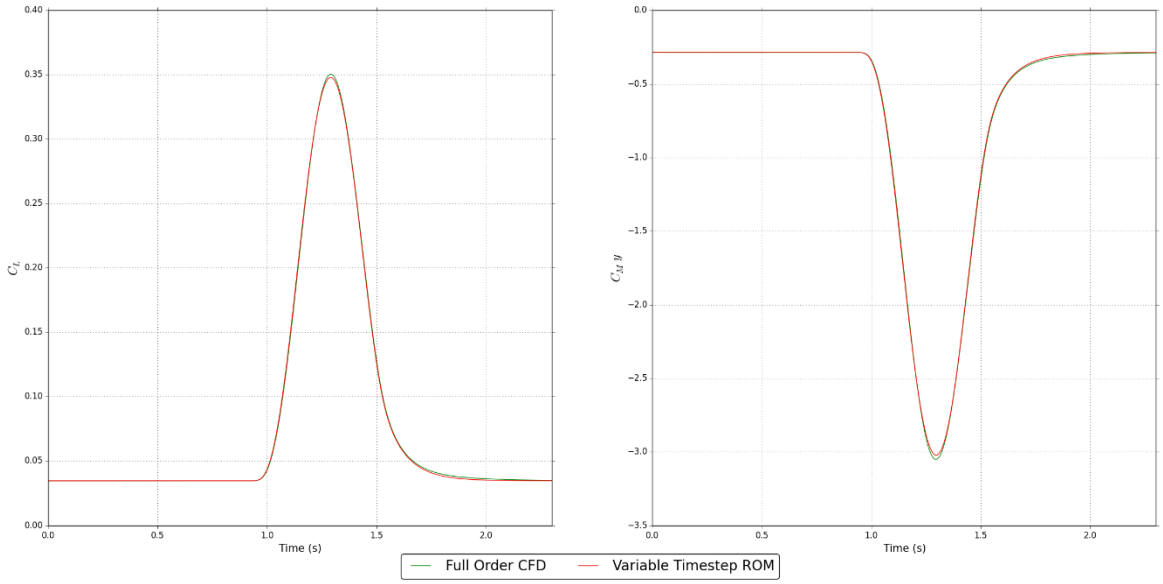

Figure 17. System response for gust case 3 calculated by the variable time-step based ROM.

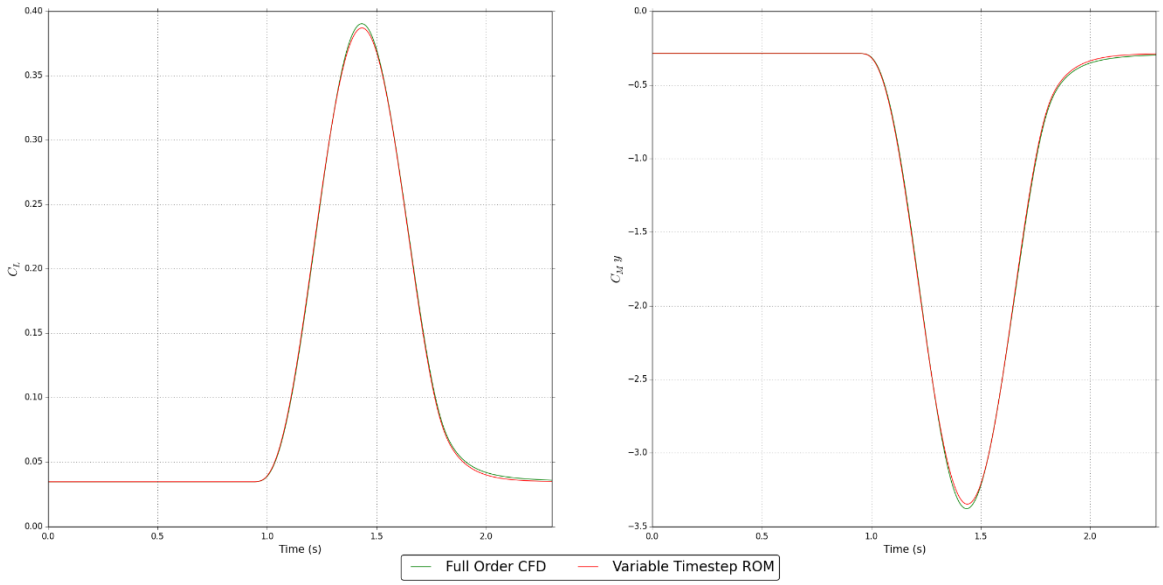

Figure 18. System response for gust case 4 calculated by the variable time-step based ROM.

E. Reduction in Sharp-Edged Gust End Point without Variable Time-Steps

The ROM based on reducing how long the sharp-edged gust simulation is required to run for shows good accuracy (see Fig. 19 to Fig. 22), with the peak response being very accurate for the short gust cases. However, the settling of the response shows an area that could benefit from further work to increase the accuracy, as well as the peak response (particularly for the pitching moment coefficient) for larger gust cases. Despite this, the overall accuracy is sufficiently adequate and the modifications result in another large saving of computational resources. By ending the sharp-edged response once the gust has cleared three mean aerodynamic chord lengths past the aft most point of the model, the number of time-steps required by the sharp-edged gust simulation was reduced from 1750 to 749. Thus this ROM resulted in a reduction in computational costs of approximately 55% from the single sharp-edged gust based ROM and approximately 75% compared to the baseline ROM.

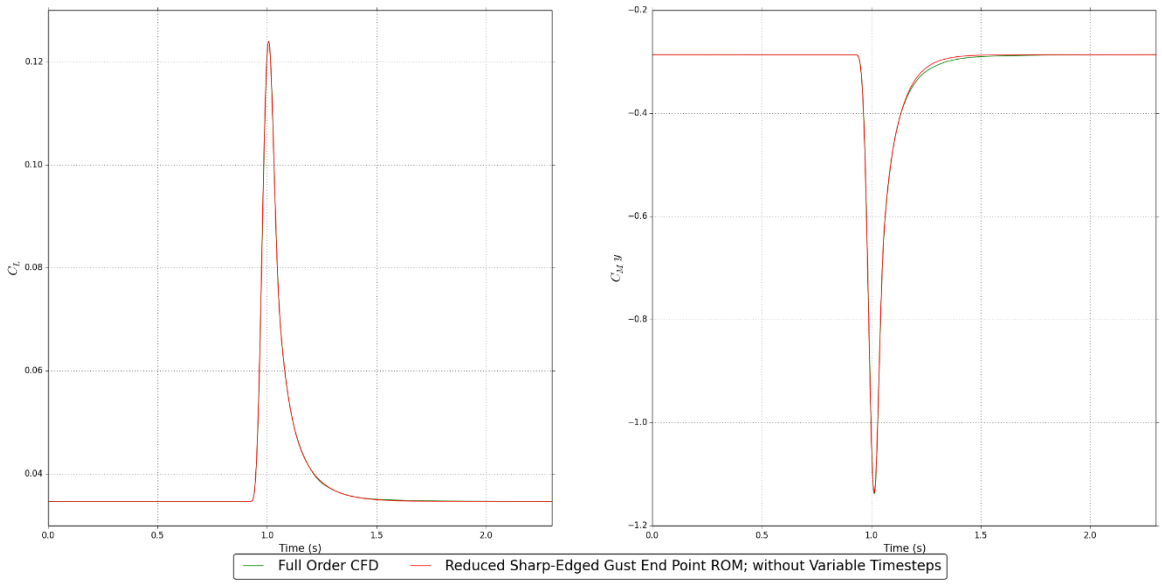

Figure 19. System response for gust case 1 calculated by ROM based on the reduction of the simulation length of the single sharp-edged gust; without variable time-steps.

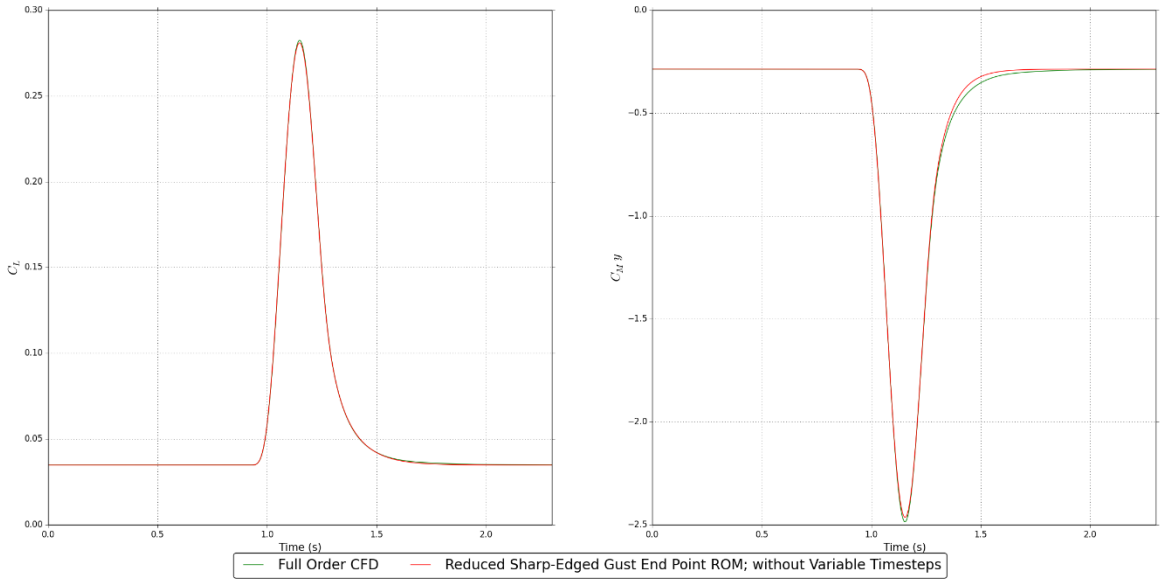

Figure 20. System response for gust case 2 calculated by ROM based on the reduction of the simulation length of the single sharp-edged gust; without variable time-steps.

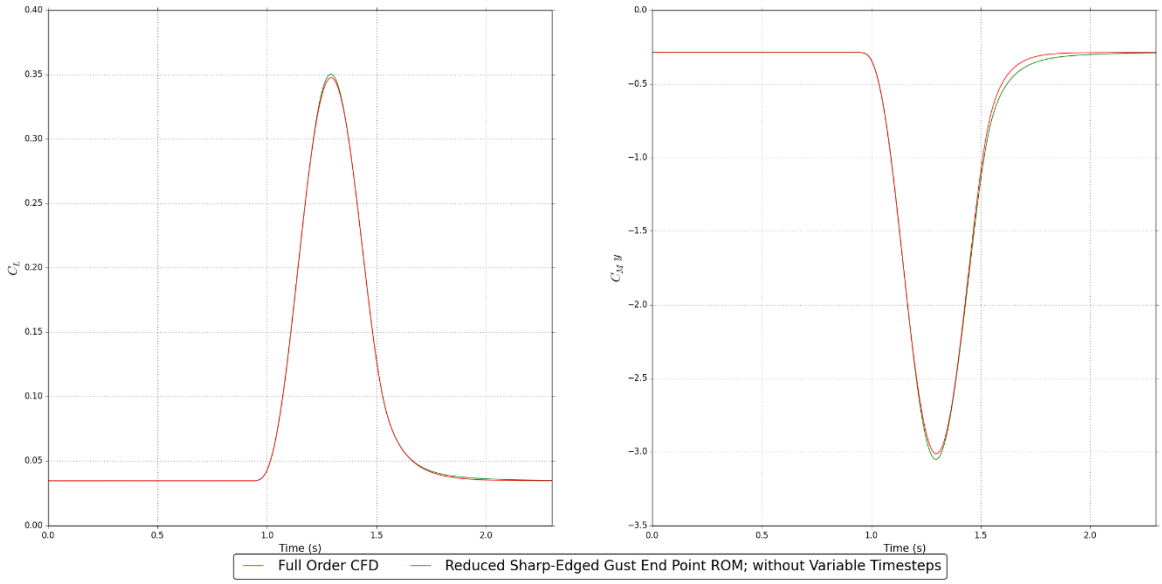

Figure 21. System response for gust case 3 calculated by ROM based on the reduction of the simulation length of the single sharp-edged gust; without variable time-steps.

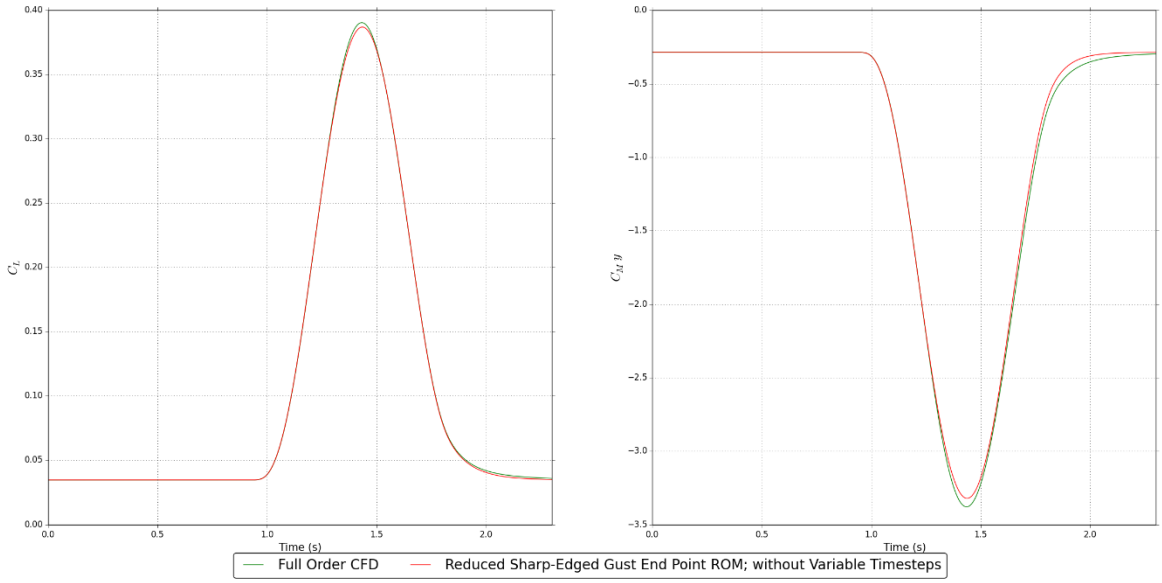

Figure 22. System response for gust case 4 calculated by ROM based on the reduction of the simulation length of the single sharp-edged gust; without variable time-steps.

F. Reduction in Sharp-Edged Gust End Point with Variable Time-Steps

It can be seen in Fig. 23 to Fig. 26 that the effects on accuracy of combining variable time-steps with the reduction in sharp-edge gust end point is minimal. For short gust lengths in particular the accuracy is very good, with only a small degradation in peak accuracy noted for the larger gust cases.

By combining the reduction in the length of the single sharp-edged gust simulation with the variable time-step size, the number of time-steps required for said simulation was reduced from the original 1750 to 173. Thus this yields a reduction in the overall computational cost of the ROM of approximately 85% when compared to the single sharp-edged gust based ROM, or a reduction of approximately 90% when compared to the baseline ROM.

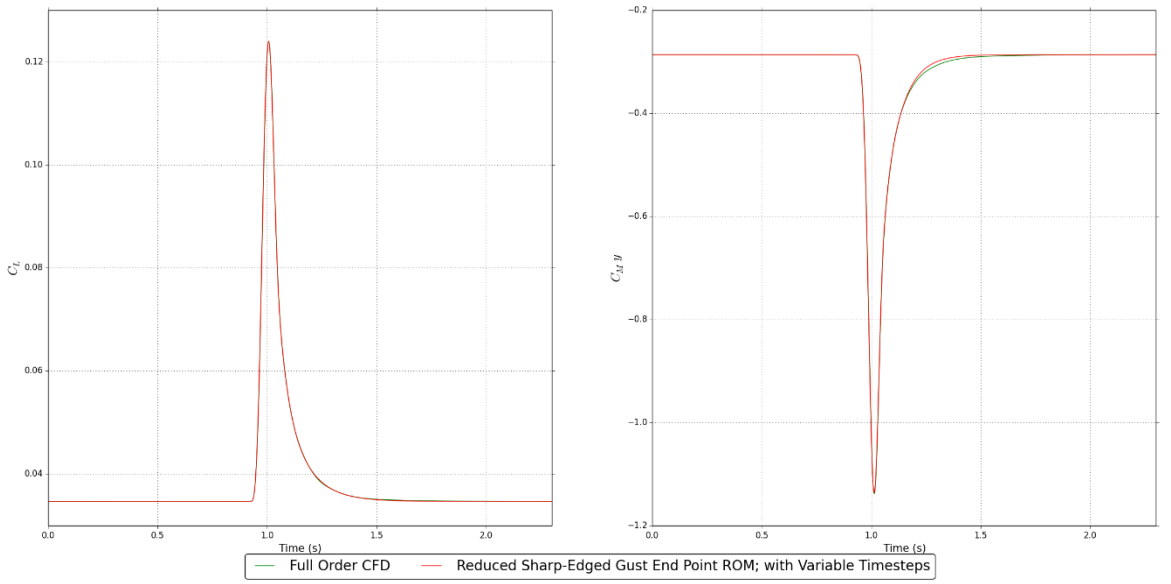

Figure 23. System response for gust case 1 calculated by ROM based on the reduction of the simulation length of the single sharp-edged gust; with variable time-steps.

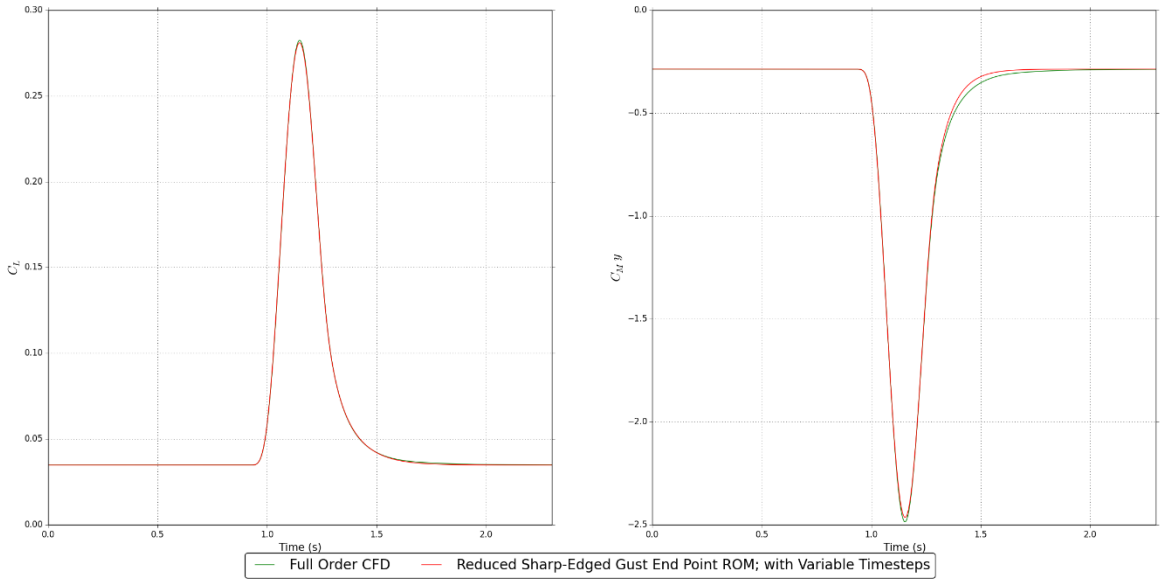

Figure 24. System response for gust case 2 calculated by ROM based on the reduction of the simulation length of the single sharp-edged gust; with variable time-steps.

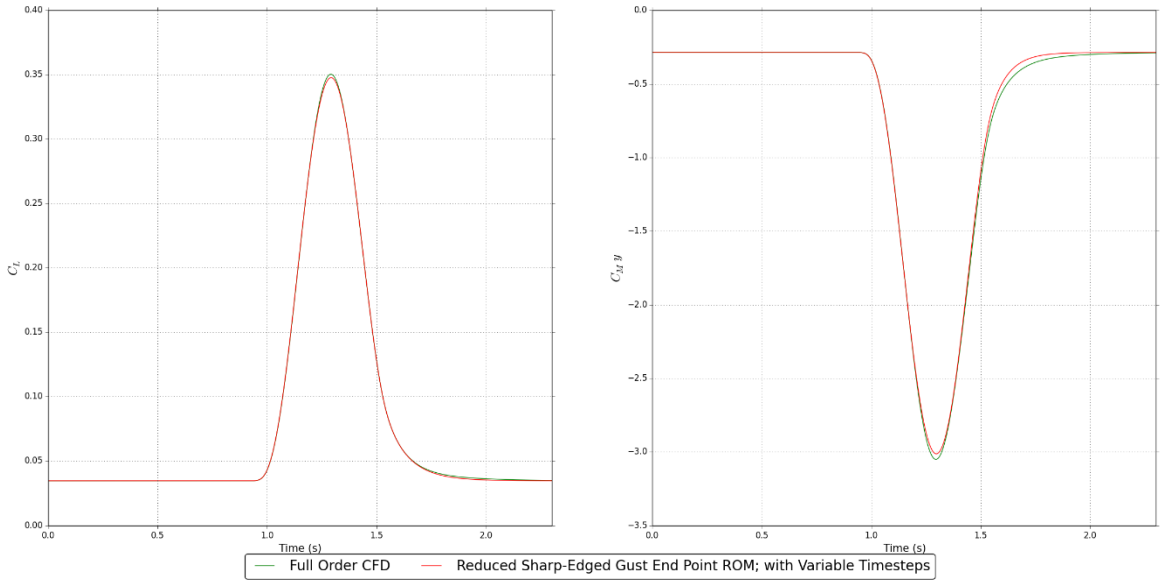

Figure 25. System response for gust case 3 calculated by ROM based on the reduction of the simulation length of the single sharp-edged gust; with variable time-steps.

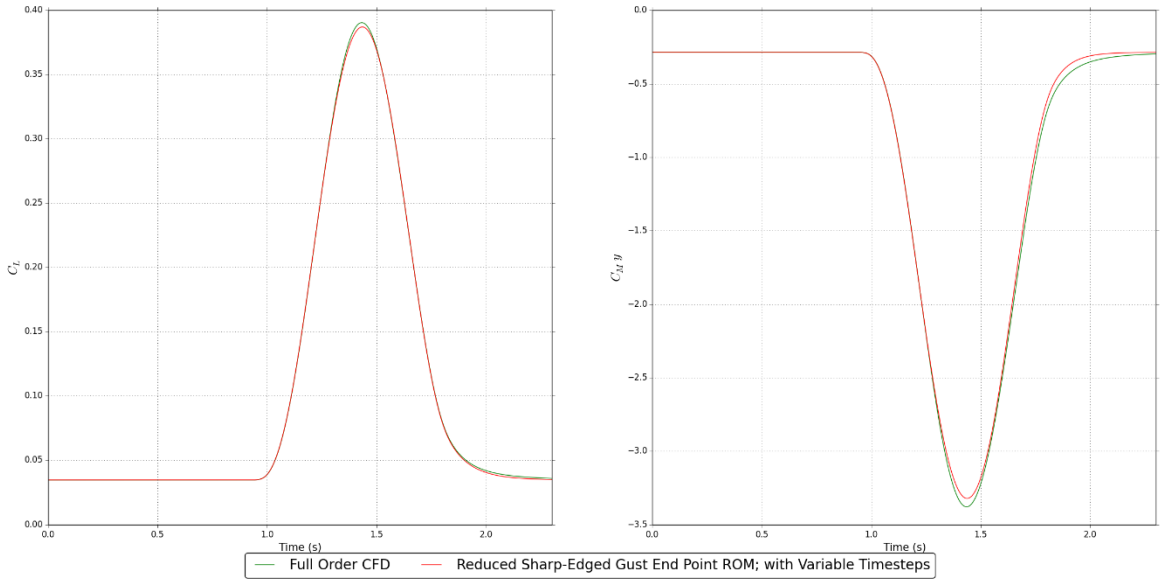

Figure 26. System response for gust case 4 calculated by ROM based on the reduction of the simulation length of the single sharp-edged gust; with variable time-steps.

V. Conclusion

The baseline method for building this ROM, as originally demonstrated by Wales *et al* [4], was shown to produce accurate results, with significant computational savings when compared with full order simulations. However, by modifying the way in which the ROM is generated, it has been shown that further savings can be made; without sacrificing accuracy.

The final ROM showed a reduction in computational costs of approximately 90% when compared to the baseline ROM; which in turn was roughly equivalent to the computational cost of one full order "1-cosine" gust simulation. As subsequent gust cases are of negligible computational cost using the ROM, for a given Mach number, then the computational savings shown in this paper open the possibility of rapid gust modelling, with countless gust lengths

at up to ten Mach numbers being able to be modelled with roughly the same computational cost required for one full order simulation to run.

There are areas that could be further explored. Firstly, the CFD simulations run in this paper were inviscid. Therefore, expanding these methods to vicious simulations may potentially offer more details on where the ROMs capture the dominant behavior, and where perhaps they can be improved. Particularly, it can be expected that a better understanding of where the ROMs fail to suitable captures the nonlinearities will be developed. This will in turn allow for the ROMs to be improved in those areas to produce a more robust tool.

In addition to the extension to viscous simulations, the outputs of the CFD simulations, and thus the ROM, could be modified to provide integrated load distributions across the wing via the discretization of the wing into strips. This would bring the outputs in line with the type of outputs often used within an industrial design loads loop. In doing so, a better understanding could potentially be developed of how well the ROM works within an important application area.

Finally, after these developments, the ROMs could be coupled with a finite element model; thus allowing for coupled aeroelastic simulations to be carried out. This would be a large undertaking, with the ROM truly being tested as it interacts with an additional solver. At this point, a clear picture should be gained on how well the process holds up when used as intended. It may be found that the interaction between the ROM and the solver cause unexpected increases in computational requirements, and if so these will need to be overcome.

Regardless of future developments, the large computational savings shown here make the modelling of multiple system responses to "1-cosine" gusts a viable prospect. This highlights the potential for them being used as a rapid modelling tool in the initial stages of aircraft design.

Acknowledgments

The authors would like to thank RCUK for the funding which has made the work laid forth in this paper possible. The authors would also like to thank Airbus Group Innovations for their support, both financial and otherwise, which has been invaluable; in particular, S. P. I. Williams would like to thank Martin Herring and Dr. John Pattinson for their ongoing support.

References

[1] Agency, E. A. S., "Certification Specifications for Large Aeroplanes; CS-25" Available: http://easa.europa.eu/system/files/dfu/CS-25_Amdt 3_19.09.07_Consolidated version.pdf.

[2] Wright, J. R., and Cooper, J. E., *Introduction to aircraft aeroelasticity and loads*, Chichester, West Sussex: Wiley, 2015.

[3] Mai, H., Neumann, J., and Hennings, H., "Gust response: a validation experiment and preliminary numcerical simulations," *15th International Forum on Aeroelasticity and Structural Dynamics*, 2011, pp. 1–20.

[4] Wales, C., Jones, D., and Gaitonde, A., "Reduced order modelling for aeroelastic aerofoil response to a gust," *51st AIAA Aerospace Sciences Meeting*, 2013, pp. 1–16.

[5] Raveh, D. E., and Zaide, A., "Numerical Simulation and Reduced-Order Modeling of Airfoil Gust Response," *AIAA Journal*, vol. 44, 2006, pp. 1826–1834.

[6] Aplevich, J. D., *The essentials of linear state-space systems*, Wiley, 2000.

[7] Juang, J.-N., and Pappa, R. S., "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," *Journal of Guidance Control and Dynamics*, vol. 8, 1985, pp. 620–627.

[8] Silva, W. A., "Application of Nonlinear Systems Theory to Transonic Unsteady Aerodynamic Responses," *Journal of Aircraft*, vol. 30, 1993, pp. 660–668.

[9] Silva, W. A., "Extension of Nonlinear systems Theory to General Frequency Unsteady Transonic Aerodynamic Responses," *34th AIAA Structures, Structural Dynamics, and Materials Conference*, Reston, VA.: AIAA, 1993, pp. 2490–2503.

[10] Silva, W. A., "Identification of Linear and Nonlinear Aerodynamic Impulse Reponses Using Digital Filter Techniques," *AIAA Atmospheric Flight Mechanics Conference*, Reston, VA.: AIAA, 1997, pp. 584–597.